Paper Title\* (use style: paper title)

Mr. Akash A. Mahajan   
*Department of Information Technology   
Rajarambapu Institute of Technology*Urun Islampur, MS, India  
akashmahajan025@gmail.com

Dr. Amol C. Adamuthe  
*Department of Information Technology   
Rajarambapu Institute of Technology*Urun Islampur, MS, India  
amol.adamuthe@gmail.com

*Abstract* — Air pollution is the ever-growing concern of the modern society because of increased industrialization and a growth in the demand of tangible products such as electronics and automobiles. Usually, industrial processes which manufacture such products pollute the ambient air via harmful gases and by-products. Sometimes, air pollution is the cause of loosely implemented environmental regulations at manufacturing yards. Air pollution is the preliminary reason for several respiratory diseases. This study, however, focuses on the forecasting of such pollutant’s concentration, especially the PM2.5 pollutant, in ambient air which can be used by governmental bodies to take needful decisions in favor of social interest. We conduct our study around the air quality data which is collected from two major metropolis of India: Delhi and Mumbai. We leverage our study on univariate time series forecasting and compare both traditional statistical methods and machine learning algorithms based on their performance in both short- and long-term predictions. Later, we benchmark our work on Urban Air Quality dataset that contains multivariate air quality data from several air monitoring systems in China. We show that machine learning model such as Random Forest Regressor will provide significantly reduced error rate for univariate time series forecasting as compared to statistical methods and certain deep learning models studied in literature.

Keywords — air quality forecasting, time series, long and short-term forecasting, machine learning, statistical methods

# Introduction

Growing industrialization and urbanization has threatened the quality of the air we breathe. Government authorities (for example, Government of India) has established air quality monitoring systems (Fig.1) in regions where chances of airborne breathing diseases are high such as in Mumbai, Delhi, Kanpur, and Kolkata etc. These monitoring devices are passive and real-time hence there is little or no way to predict the air quality which is potentially beneficial for government to spread awareness or take important decisions associated with public interest.

Map

Description automatically generated

Fig.1 Air monitoring stations across New Delhi

Air quality is broadly affected by certain pollutants such as Particulate Matters – PM2.5 and PM10, SO2, CO, NO2 etc. [1] Fortunately, air quality can be forecasted based on historical data collected from these air quality monitoring systems. Various techniques like statistics, Machine Learning, and deep learning can be applied to these available data to predict the air quality [2, 3, 4, 5, 6] and more specifically the concentration of PM2.5 in air for certain cities around India. We use publicly available historical data for the PM2.5 concentration in the metropolitan cities like Delhi and Mumbai. Though several factors affect the quality of the air [1], we do not discuss them here because of the scope of the study.

In this study, we forecast univariate time series and produce a comparison of the statistical methods such as ARIMA [7], its variants [8] and several machine learning regression methods such as support vector machine, XGBoost, Decision Trees, and Random Forest [9]. The aim of this study is to investigate the capabilities of these methods when forecasting over a short-term and long-term period. In the rest of this paper, section II will introduce the related work in the air quality forecasting. Section III gives the overview of the methodology, statistical and machine learning models used for comparison. Section IV briefly describes the conducted experiments, evaluations, analysis, and the associated results. We finally conclude our work in section V and provide few suggestions of future work in last section of this paper.

# Related work

Time series analysis is the important field of study associated with analysis of a given temporal data to find out useful patterns, trends, and correlations [10]. Usually, the found patterns are used for forecasting and prediction of a certain events in the future. Traditionally, time series forecasting is carried out with the help of statistical methods such as Moving Averages, Autoregressive Moving Average (ARMA, ARIMA), and SARIMA [8, 11]. Time series forecasting methods based on statistics and machine learning have been used by various researchers for forecasting gold price [12], crop yield [13], stock prices [14], and electricity load [15], and air pollution. Deep learning models such as CNN, RNN, and LSTMs are proved reliable on several univariate and multivariate benchmarks for certain forecasting applications listed above [16]. Deep learning models are employed for time series forecasting over the statistical methods as deep learning methods are good at extracting special patterns in the data which is probably not possible through statistical methods. For example, [3] proposed a deep Recurrent Neural Network for forecasting the air pollution through time series data, [17] proposed a LSTM model for predicting the stock prices of Indian market by tuning some special parameters of LSTM. In some cases, hybrid neural network models such as mentioned in [2] and [18] improves the capabilities of the neural network in terms of its training time and forecasting performance. Deep learning models are good at performance but consumes valuable computing resources, time, and could take even weeks to train. Deep learning models are trained on larger dataset hence are capable to capture trends and patterns in data. However, a relatively smaller dataset is not sufficient for training a complex neural network model. As mentioned, because of huge consumption of computing resources it is not desired to train a neural network model on smaller dataset. That is when machine learning algorithms are proven to be strong for certain applications. For example, [19] proposes a nearest neighbor model for forecasting the financial time series, [4] uses gradient boosted machine learning algorithms for forecasting the air quality in Taiwan.

In our work, we compare several statistical methods and machine learning methods which are popular for time series forecasting. The time series data includes air quality data from New Delhi and Mumbai. The achieved results are compared with few deep learning models while benchmarking it with urban air quality dataset.

# Methodology

In this section, we will briefly outline the statistical and machine learning methods used for comparison, the dataset, the experimental setup, and metrics used for evaluation.

## ARIMA [7]

ARIMA stands for Autoregressive Integrated Moving Average, also called the Box-Jenkins method. It is formed with autoregressive (AR), Integrated (I), and Moving Average (MA) terms. An *autoregressive* process is the process by which future values are predicted with help of several past values of the same attribute. The *integrated* term holds value of difference between the current observation and the past observation of same attribute. And the *moving average* is the weighted moving average of previous forecasting errors. ARIMA has three different parameters Viz., p, d, and q which are lag order, degree of differencing, and moving average window size, respectively. These values jointly called as order of ARIMA (p, d, q).

On the other hand, there is a seasonal variant of ARIMA which involves a seasonal order, also called SARIMA [20]. With the help of SARIMA, we can model the seasonal component present in the time series data. It is extended version of the original ARIMA process hence involves original p, d, q terms and new terms for seasonal order added as P, D, Q, and m where P, D, Q are similar terms as ARIMA but with seasonal component considered. m stands for the period of seasonality (Usually 12 month). The SARIMA order usually represented as *ARIMA (p, d, q) x (P, D, Q, m).*

We use four variants of ARIMA:

* First order autoregressive ARIMA of order (1, 0, 0)
* Damped-trend linear exponential smoothing ARIMA of order (1, 1, 2)
* Differenced first order autoregressive ARIMA of order (1, 1, 0), and
* Seasonal ARIMA of order (1, 1, 0) x (0, 1, 1, 12)

## Machine learning methods

We compare popular machine learning regression model described as follows:

### Support Vector Regression (SVR): Support Vector Regressor is a SVM regressor which build a mapping function that will map the given data point to a real number based on the training data. It uses a kernel method to map the values to its appropriate class or numerical value. For our experiment, we use SVR with Radial Basis Function (RBF) kernel.

### XG Boost Regression (XGBR): XGBoost regressor is the ensemble learning approach. It is constructed on decision tree models. Instead of creating a model of multiple trees at one instance, one tree is added to the model ensemble at a time while fitting them on the data using any loss function and gradient descent optimization.

### Decision Tree Regression (DTR): The goal of implementing a Decision Tree Regressor is to create a regression model based on inffered rules on features found in the dataset. A decision tree is capable of approximating a sine curve based on the decisions.

### Random Forest Regression (RFR): A Random Forest algorithm is also an ensemble of decision trees. It involves several decision trees (that’s why the name “forest”). Each tree will work on different feature set and will predict an appropriate value based on its learning. Several predicted values are averaged for better prediction.

## Dataset

We compare above mentioned methods based on the air quality data collected from three different dataset Viz., Delhi AQI dataset, Mumbai AQI dataset [21], and Urban Air Quality dataset [2]. The PM2.5 concentration values in all datasets are highly variable and non-linear (See Fig. 2). Some basic information of these dataset provided below:

1. basic information about datasets

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset** | **Number of records** | **Type** | **Missing records** |
| Delhi AQI | 2687 | Univariate | 0 |
| Mumbai AQI | 2684 | Univariate | 0 |
| Urban Air Quality dataset | 7552 | Univariate | 164 |

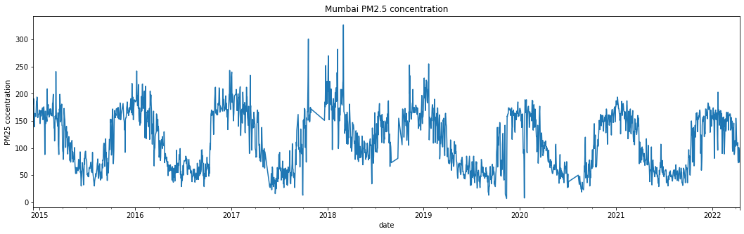


Fig.2 PM2.5 concentration over the time for Mumbai AQI dataset

Delhi AQI and Mumbai AQI datasets includes daily PM2.5 concentration score from December 2014 to April 2022. Urban air quality dataset has air quality data from 437 different air monitoring stations in China for several air pollutants. The data is recorded on hourly basis. It has more than 2.8 million records but for our experimentation we only use PM2.5 concentration records of the station number 1001.

### Treatment of missing values: In urban air quality dataset, 164 entries for PM2.5 concentration were missing (almost 2.17% of whole dataset). We fill the missing values of PM2.5 attribute for urban air quality dataset with the median value of the attribute.

## Experimental setup

We use Python’s statistics model library to implement ARIMA and its variants. To implement machine learning algorithms, we use Scikit-learn library of Python. All code is tested using Jupyter Notebook. Computing facility used is Dell Precision Tower 3640 workstation with Intel® 10th generation i9 processor, 128GB RAM and 16GB Nvidia Quadro RTX 5000 graphics card. A special Python function for preprocessing the datasets to prepare them for the machine learning models is described below:

### Creating the data matrix: We represent the univariate data into a tabular matrix. The function create\_matrix has two mandatory parameters:

#### data: The data to be preprocessed represented with a NumPy array.

#### look\_back: The number of records to consider before predicting the next value (also called order of lag). The default is 5.

The flow of the *create\_matrix* function illustrated in Fig. 2.

Table

Description automatically generated

Fig. 3 Illustration of create\_matrix function

We predict the values in two terms: short term and long term. Short term prediction period is of 30 days whereas long term prediction is of 250 days. The prediction errors are evaluated using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). The equation for the same are given below:

(1)

(2)

# Result and discussion

This section will briefly elaborate the dataset-wise conducted experiments and associated results. Section A and Section B will cover the short- and long-term forecasting on Delhi AQI dataset, respectively. Section C and D will cover the short- and long-term forecasting on Mumbai dataset, respectively. Section E will show overall comparison of statistical methods and machine learning methods on both datasets. In the section F, we will benchmark best worked model on Urban Air Quality dataset. In the rest of part, we will show the comparison between best worked model and two deep learning models on Urban Air Quality dataset in literature.

## Analysis of Short term forecasting on Delhi AQI dataset

Table II and table III shows the short-term forecasting errors for Delhi dataset for statistical and machine learning methods, respectively.

According to figures in table II, damped-trend linear exponential smoothing ARIMA (method 2) outperformed other methods. However, certain methods such as method 1 and 3 shown sensitivity for a specific forecasting period. For example, method 1 is sensitive to period of 5 to 15 in terms of MAE. Values marked in bold are the lowest errors for a certain forecasting period. By considering the performance in all three periods, method 2 outperformed.

1. Short term forecast results of statistical methods on Delhi AQI dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Model** | **Period (days)** | **MAE** | **RMSE** |
| 1 | First order autoregressive ARIMA | 1~5 | 7.82 | 10.14 |
| 5~15 | **8.20** | 10.28 |
| 15~30 | 13.24 | 15.17 |
| **2** | **Damped-trend linear exponential smoothing ARIMA** | 1~5 | 7.43 | 8.97 |
| 5~15 | 8.45 | 9.94 |
| 15~30 | **9.87** | 12.00 |
| 3 | Differenced first order autoregressive ARIMA | 1~5 | 7.50 | 8.93 |
| 5~15 | 9.29 | 11.65 |
| 15~30 | 13.66 | 15.56 |
| 4 | SARIMA | 1~5 | **7.29** | 8.63 |
| 5~15 | 8.49 | 10.72 |
| 15~30 | 13.44 | 15.24 |

For machine learning methods, Support Vector Regressor performed the best. Though XGBoost regressor is sensitive to the period of 1-5 days forecasting, considering the performance in all three periods of short-term forecast Support Vector Regressor was best among all compared. (Fig. 4)

1. Short term forecast results of machine learning methods on Delhi AQI datasett

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Model** | **Period (days)** | **MAE** | **RMSE** |
| 1 | Support Vector Regressor | 1~5 | **9.15** | 10.17 |
| 5~15 | **9.24** | 10.67 |
| 15~30 | **9.95** | 12.68 |
| 2 | XGBoost Regressor | 1~5 | 8.96 | 10.59 |
| 5~15 | 9.61 | 12.38 |
| 15~30 | 13.61 | 18.28 |
| 3 | Decision Tree Regressor | 1~5 | 14.60 | 16.80 |
| 5~15 | 19.70 | 24.85 |
| 15~30 | 23.84 | 30.59 |
| 4 | Random Forest Regressor | 1~5 | 9.44 | 10.66 |
| 5~15 | 9.59 | 10.73 |
| 15~30 | 12.12 | 15.58 |

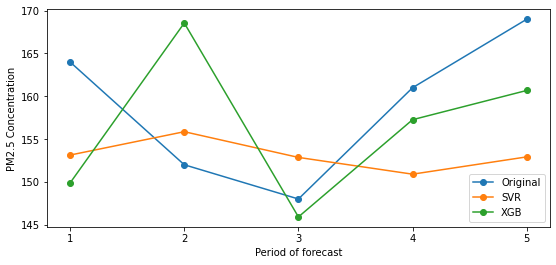


Fig. 4 Support Vector Regressor Vs XGBoost Regressor on 1-5 days forecasting on Delhi AQI dataset

We compared the Random Forest Regressor with Support Vector Regressor for 5-15 days forecasting period, as both methods have nearly equal MAE error rate. (See Fig. 5)

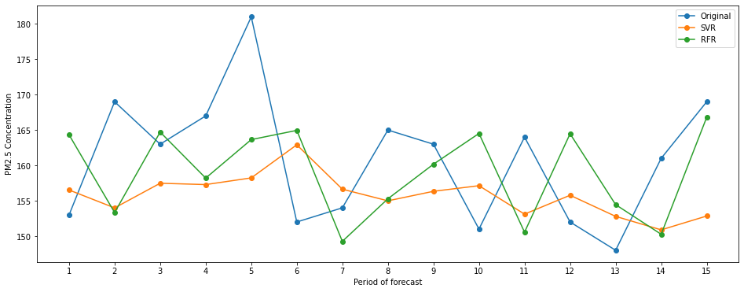


Fig. 5 Support Vector Regressor Vs Random Forest Regressor on 5-15 days forecasting on Delhi AQI dataset

Well, Damped-trend linear exponential smoothing ARIMA (Table II, method 2) still performed better than the of Support Vector Regressor (Table III, method 1).

## Analysis of long term forecasting on Delhi AQI dataset

Table IV and V shows the long-term forecasting errors on the Delhi AQI dataset of statistical and machine learning methods, respectively.

1. Long term forecast results of statistical methods on Delhi AQI datasett

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Model** | **Period (days)** | **MAE** | **RMSE** |
| 1 | **First order autoregressive ARIMA** | 30~60 | 17.03 | 23.95 |
| 60~100 | **28.27** | 43.05 |
| 100~250 | **54.36** | 73.94 |
| 2 | Damped-trend linear exponential smoothing ARIMA | 30~60 | 17.11 | 24.05 |
| 60~100 | 51.57 | 56.55 |
| 100~250 | 89.21 | 112.82 |
| 3 | Differenced first order autoregressive ARIMA | 30~60 | 16.04 | 23.09 |
| 60~100 | 47.36 | 60.17 |
| 100~250 | 97.52 | 121.39 |
| 4 | SARIMA | 30~60 | **14.63** | 21.63 |
| 60~100 | 49.47 | 61.51 |
| 100~250 | 104.67 | 128.15 |

Statistical methods cannot perform well when the forecasting period is long. For example, method 2 when used for short-term forecasting has very low error rate as compared to long-term forecast. As compared with other methods, first order autoregressive ARIMA model (method 1) performed better but results are not desirable.

1. Long term forecast results of machine learning methods on Delhi AQI datasett

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Model** | **Period (days)** | **MAE** | **RMSE** |
| 1 | Support Vector Regressor | 30~60 | **14.54** | 20.61 |
| 60~100 | 21.58 | 31.15 |
| 100~250 | 30.53 | 44.48 |
| 2 | XGBoost Regressor | 30~60 | 16.61 | 22.42 |
| 60~100 | 22.10 | 29.72 |
| 100~250 | 29.17 | 42.50 |
| 3 | Decision Tree Regressor | 30~60 | 25.01 | 31.70 |
| 60~100 | 38.78 | 62.68 |
| 100~250 | 40.19 | 58.55 |
| 4 | Random Forest Regressor | 30~60 | 15.28 | 20.72 |
| 60~100 | **21.22** | 31.26 |
| 100~250 | **26.46** | 39.29 |

Random Forest Regressor, among the other methods worked best for long term predictions. And even it performed better than first order autoregressive model, for example, reduced RMSE by 49.08% for the forecast period of 100 to 250 days. (See Fig. 6.a and 6.b)

## Analysis of short term forecasting on Mumbai Dataset

1. Short term forecast results of statistical methods on Mumbai AQI datasett

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Model** | **Period (days)** | **MAE** | **RMSE** |
| 1 | First order autoregressive ARIMA | 1~5 | **9.47** | 11.04 |
| 5~15 | 19.27 | 25.05 |
| 15~30 | **21.45** | 25.11 |
| 2 | Damped-trend linear exponential smoothing ARIMA | 1~5 | 14.61 | 16.98 |
| 5~15 | 21.14 | 25.66 |
| 15~30 | 37.70 | 44.03 |
| 3 | Differenced first order autoregressive ARIMA | 1~5 | 10.78 | 15.12 |
| 5~15 | **17.47** | 23.31 |
| 15~30 | 34.55 | 40.58 |
| 4 | SARIMA | 1~5 | 9.80 | 13.44 |
| 5~15 | 17.58 | 23.29 |
| 15~30 | 34.40 | 40.15 |

Fig. 6 Random Forest Regressor Vs Support Vector Machine performance for a) 100 days forecast and b) 250 days forecast on Delhi dataset

Table VI and VII outlines the short-term forecasting errors obtained on Mumbai AQI dataset for statistical and machine learning methods, respectively. In table VI, first order autoregressive ARIMA outperformed other statistical models. In case of 5 to 15 days forecast, method 3 is sensitive to the Mumbai AQI dataset.

1. Short term forecast results of machine learning methods on Mumbai AQI dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Model** | **Period (days)** | **MAE** | **RMSE** |
| 1 | Support Vector Regressor | 1~5 | 13.87 | 15.98 |
| 5~15 | 14.54 | 18.83 |
| 15~30 | 15.98 | 21.56 |
| 2 | XGBoost Regressor | 1~5 | **7.27** | 8.08 |
| 5~15 | 15.54 | 20.37 |
| 15~30 | 12.98 | 18.83 |
| 3 | Decision Tree Regressor | 1~5 | 10.50 | 12.51 |
| 5~15 | 16.00 | 19.70 |
| 15~30 | 17.97 | 22.98 |
| 4 | **Random Forest Regressor** | 1~5 | 11.54 | 12.17 |
| 5~15 | **11.73** | 17.31 |
| 15~30 | **12.93** | 18.78 |

According to table VII, Random Forest Regressor worked best (See Fig. 7), where XGBoost Regressor is sensitive on the Mumbai dataset when the forecasting period is 1 to 5 days.

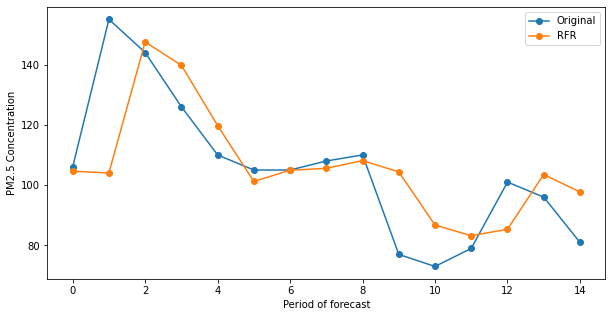


Fig. 7. 5 to 15 days forecast using Random Forest Regressor on Mumbai AQI dataset.

## Analysis of long term forecasting on Mumbai Dataset

Table VIII and IX shows the long-term forecasting errors on the Mumbai AQI dataset for the statistical methods and machine learning methods, respectively.

1. Long term forecast results of Statistical methods on Mumbai AQI dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Model** | **Period (days)** | **MAE** | **RMSE** |
| 1 | **First order autoregressive ARIMA** | 30~60 | **25.77** | 29.37 |
| 60~100 | **27.65** | 31.81 |
| 100~250 | **37.81** | 41.90 |
| 2 | Damped-trend linear exponential smoothing ARIMA | 30~60 | 25.27 | 30.80 |
| 60~100 | 30.03 | 40.54 |
| 100~250 | 67.60 | 79.73 |
| 3 | Differenced first order autoregressive ARIMA | 30~60 | 24.56 | 29.14 |
| 60~100 | 38.22 | 47.45 |
| 100~250 | 63.42 | 75.14 |
| 4 | SARIMA | 30~60 | 24.48 | 28.99 |
| 60~100 | 38.08 | 47.38 |
| 100~250 | 68.19 | 80.18 |

1. Long term forecast results of machine learning methods on Mumbai AQI dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Model** | **Period (days)** | **MAE** | **RMSE** |
| 1 | Support Vector Regressor | 30~60 | 15.13 | 20.03 |
| 60~100 | 16.44 | 22.43 |
| 100~250 | 15.21 | 20.69 |
| 2 | XGBoost Regressor | 30~60 | 13.65 | 18.81 |
| 60~100 | 15.62 | 21.66 |
| 100~250 | 13.74 | 19.52 |
| 3 | Decision Tree Regressor | 30~60 | 20.16 | 27.80 |
| 60~100 | 21.80 | 28.06 |
| 100~250 | 18.15 | 24.31 |
| 4 | **Random Forest Regressor** | 30~60 | **13.25** | 18.16 |
| 60~100 | **15.05** | 21.01 |
| 100~250 | **13.04** | 18.68 |

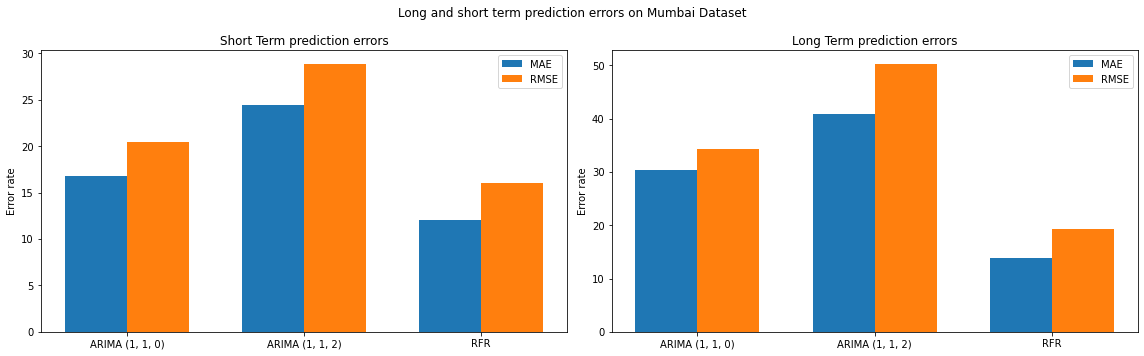
Random Forest Regressor still performed better as compared to other methods.

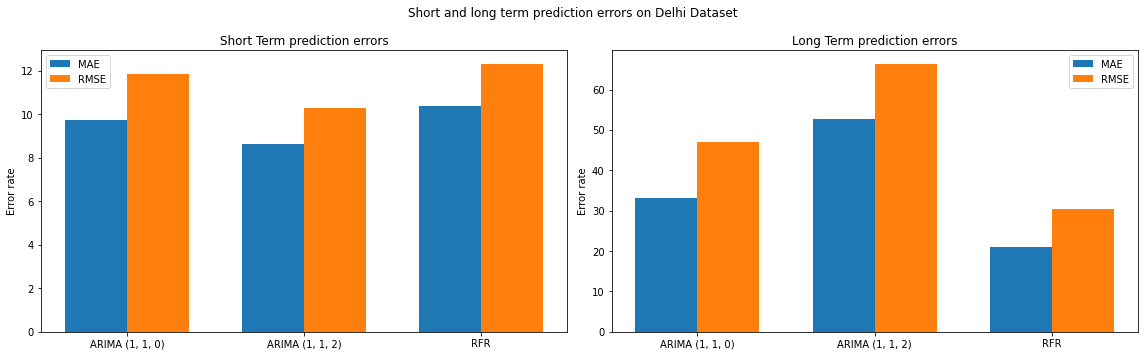
## Collective results of statistical and machine learning methods on Delhi AQI and Mumbai AQI datasets.

After a thorough comparison between several statistical and machine learning methods, we found three such models which have repeatedly outperformed other models Viz., First order autoregressive ARIMA, Damped-trend linear exponential smoothing ARIMA, and the Random Forest Regressor. Table X shows the average error rates of each of the above-mentioned method for short-term and long-term forecasting. Overall, in all periods Random Forest Regressor performed the best as compared to remaining two statistical methods. See Fig. 8 for illustration of collective results.

1. Comparison of averaged short and long term error rates of top performing methods on Delhi AQI and Mumbai AQI datasets

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dataset** | **Model** | **Short Term** | | **Long Term** | |
| **Avg. MAE** | **Avg. RMSE** | **Avg. MAE** | **Avg. RMSE** |
| Delhi AQI | First order autoregressive ARIMA | 9.75 | 11.86 | 33.22 | 46.98 |
| Damped-trend linear exponential smoothing ARIMA | **8.62** | 10.30 | 52.63 | 64.47 |
| **Random Forest Regressor** | 10.38 | 12.32 | **20.98** | 30.42 |
| Mumbai AQI | First order autoregressive ARIMA | 16.73 | 20.40 | 30.41 | 34.36 |
| Damped-trend linear exponential smoothing ARIMA | 24.48 | 28.89 | 40.96 | 50.35 |
| **Random Forest Regressor** | **12.07** | 16.08 | **13.78** | 19.28 |





## Benchmarking on Urban Air Quality dataset

Table XI shows the forecasting errors of Random Forest Regressor when applied on short-term and long-term forecasting on Urban Air quality dataset. Note that, Urban Air Quality dataset includes hourly PM2.5 Concentration data.

1. Random Forest Regressor forecasting results on Urban Air QUality Dataset (From 1 hour to 250 hours forecast)

|  |  |  |
| --- | --- | --- |
| **Period (hours)** | **MAE** | **RMSE** |
| 1~5 | 9.35 | 10.49 |
| 5~15 | 13.91 | 21.84 |
| 15~30 | 10.73 | 17.17 |
| 30~60 | 8.33 | 13.23 |
| 60~100 | 9.14 | 13.51 |
| 100~250 | 11.06 | 15.51 |

A least significant growth in error rates (both MAE and RMSE) is observed as increase in forecasting period. For example, for the period of 1 to 5 hours, the MAE observed is 9.35 and for the period of 100 to 250 hours, it is 11.06.

We compared the Random Forest Regressor with the smallest forecasting error rates of two deep learning models present in [2] and [6] as shown in table XII. Please note that, the MAE for Random Forest Regressor is averaged. For example. for short term period, the average of MAE for periods 1~5, 5~15, and 15~30 is taken. Comparison shown a significant difference between forecasting error rates of deep learning models and the Random Forest Regressor used for Urban Air Quality dataset. (See Table XII)

1. Comparison of deep learning models with random forest regressor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Random Forest Regressor (Long Term)** | **Random Forest Regressor (Short term)** | **[6]** | **DAQFF [2]** |
| MAE | 9.51 | 11.33 | 23.70 | 25.01 |

# Conclusion

We compare several statistical and machine learning methods used for time series forecasting to forecast the air quality in terms of PM2.5 concentration in the air. We take three different datasets for forecasting: Delhi AQI, Mumbai AQI, and the Urban Air Quality dataset. After thorough comparison between various forecasting models, we observe that First order autoregressive ARIMA, Damped-trend linear exponential smoothing ARIMA, and the Random Forest Regressor have performed the best. After interpretation of collective results, we observe that Random Forest Regressor excels in both kind of forecasting. Further, we forecast the PM2.5 concentration using the Random Forest Regressor on Urban Air Quality dataset for both short term and long terms periods. Later, we compare the averaged short-term and long-term error rates with best error rates from literature. We show that Random Forest Regressor will outperform the deep learning models from the literature. In the future, we can hybridize the Random Forest Regressor with little or huge customization with deep learning models to get state-of-the-art results not only on time series analysis but also for the various applications like classification etc.

##### references

[1] Sharma, R., Kumar, R., Sharma, D. K., Son, L. H., Priyadarshini, I., Pham, B. T., ... & Rai, S. (2019). Inferring air pollution from air quality index by different geographical areas: case study in India. Air Quality, Atmosphere & Health, 12(11), 1347-1357.

[2] Du, S., Li, T., Yang, Y., & Horng, S. J. (2019). Deep air quality forecasting using hybrid deep learning framework. IEEE Transactions on Knowledge and Data Engineering, 33(6), 2412-2424.

[3] Ong, B. T., Sugiura, K., & Zettsu, K. (2016). Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting PM2. 5. Neural Computing and Applications, 27(6), 1553-1566.

[4] Lee, M., Lin, L., Chen, C. Y., Tsao, Y., Yao, T. H., Fei, M. H., & Fang, S. H. (2020). Forecasting air quality in Taiwan by using machine learning. Scientific reports, 10(1), 1-13.

[5] Freeman, B. S., Taylor, G., Gharabaghi, B., & Thé, J. (2018). Forecasting air quality time series using deep learning. Journal of the Air & Waste Management Association, 68(8), 866-886.

[6] Zheng, Y., Yi, X., Li, M., Li, R., Shan, Z., Chang, E., & Li, T. (2015, August). Forecasting fine-grained air quality based on big data. In Proceedings of the 21st ACM SIGKDD international conference on knowledge discovery and data mining (pp. 2267-2276).

[7] ARIMA: Box, G. E. (1970). GM Jenkins Time Series Analysis: Forecasting and Control. San Francisco, Holdan-Day.

[8] Divisekara, R. W., Jayasinghe, G. J. M. S. R., & Kumari, K. W. S. N. (2021). Forecasting the red lentils commodity market price using SARIMA models. SN Business & Economics, 1(1), 1-13.

[9] Mahesh, B. (2020). Machine learning algorithms-a review. International Journal of Science and Research (IJSR). [Internet], 9, 381-386.

[10] Chatfield, C. (2000). Time-series forecasting. Chapman and Hall/CRC.

[11] Nelson, B. K. (1998). Time series analysis using autoregressive integrated moving average (ARIMA) models. Academic emergency medicine, 5(7), 739-744.

[12] Manjula, K. A., & Karthikeyan, P. (2019, April). Gold price prediction using ensemble-based machine learning techniques. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 1360-1364). IEEE.

[13] SARIMA application - Reddy, P., & Sureshbabu, A. (2019, June). An applied time series forecasting model for yield prediction of agricultural crop. In International Conference on Soft Computing and Signal Processing (pp. 177-187). Springer, Singapore.

[14] Leung, C. K. S., MacKinnon, R. K., & Wang, Y. (2014, July). A machine learning approach for stock price prediction. In Proceedings of the 18th International Database Engineering & Applications Symposium (pp. 274-277).

[15] Cai, M., Pipattanasomporn, M., & Rahman, S. (2019). Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques. Applied energy, 236, 1078-1088.

[16] Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning: a survey. Philosophical Transactions of the Royal Society A, 379(2194), 20200209.

[17] Yadav, A., Jha, C. K., & Sharan, A. (2020). Optimizing LSTM for time series prediction in Indian stock market. Procedia Computer Science, 167, 2091-2100.

[18] Livieris, I. E., Pintelas, E., & Pintelas, P. (2020). A CNN–LSTM model for gold price time-series forecasting. Neural computing and applications, 32(23), 17351-17360.

[19] Ban, T., Zhang, R., Pang, S., Sarrafzadeh, A., & Inoue, D. (2013, November). Referential knn regression for financial time series forecasting. In International Conference on Neural Information Processing (pp. 601-608). Springer, Berlin, Heidelberg.

[20] Hipel, K. W., McLeod, A. I., & Lennox, W. C. (1977). Advances in Box‐Jenkins modeling: 1. Model construction. Water Resources Research, 13(3), 567-575.

[21] Air quality Data used for experiments is available at: <https://in.usembassy.gov/embassy-consulates/new-delhi/air-quality-data/>